

TAMER: Tree-Aware Transformer for Handwritten Mathematical Expression Recognition

Jianhua Zhu, Wenqi Zhao, Yu Li, Xingjian Hu, Liangcai Gao*

Wangxuan Institute of Computer Technology, Peking University, Beijing, China
 zhujianhuapku@pku.edu.cn, {wenqizhao, liyu}@stu.pku.edu.cn, {huxingjian, gaoliangcai}@pku.edu.cn

Abstract

Handwritten Mathematical Expression Recognition (HMER) has extensive applications in automated grading and office automation. However, existing sequence-based decoding methods, which directly predict LaTeX sequences, struggle to understand and model the inherent tree structure of LaTeX and often fail to ensure syntactic correctness in the decoded results. To address these challenges, we propose a novel model named TAMER (Tree-Aware Transformer) for handwritten mathematical expression recognition. TAMER introduces an innovative Tree-aware Module while maintaining the flexibility and efficient training of Transformer. TAMER combines the advantages of both sequence decoding and tree decoding models by jointly optimizing sequence prediction and tree structure prediction tasks, which enhances the model’s understanding and generalization of complex mathematical expression structures. During inference, TAMER employs a Tree Structure Prediction Scoring Mechanism to improve the structural validity of the generated LaTeX sequences. Experimental results on CROHME datasets demonstrate that TAMER outperforms traditional sequence decoding and tree decoding models, especially in handling complex mathematical structures, achieving state-of-the-art (SOTA) performance.

Code — <https://github.com/qingzhenduyu/TAMER/>

Introduction

The task of Handwritten Mathematical Expression Recognition (HMER) holds a unique and important place in the field of Optical Character Recognition (OCR). Compared to traditional OCR tasks, it faces more complex challenges. Traditional character recognition models typically focus on recognizing linearly arranged text sequences, whereas the structure of handwritten mathematical expressions is not one-dimensional and linear but exhibits multi-layered and highly structured characteristics. For example, mathematical symbols such as fractions, square roots, and integrals involve not only basic characters but also complex structures like subscripts, superscripts, and nested symbols, all of which greatly increase the difficulty of recognition. Therefore, a

*Corresponding author.

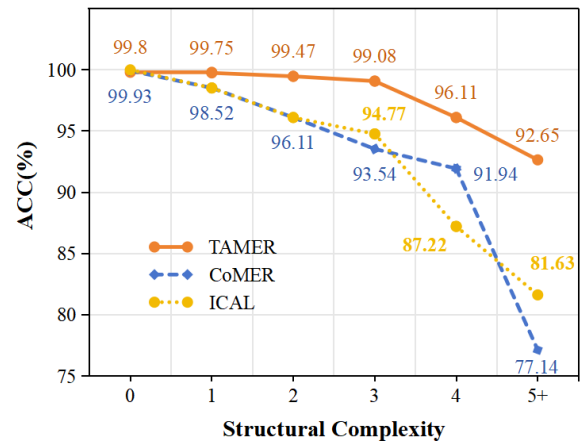


Figure 1: **Bracket Matching Accuracy under different structural complexities on CROHME 2014 (in %).** TAMER maintains a bracket matching accuracy of over 92% across all levels of structural complexity, significantly outperforming CoMER and ICAL.

core challenge in recognizing handwritten mathematical expressions is how to effectively encode the structural information of the expressions within the model. This requires the model to not only accurately identify characters in the image but more importantly, to construct the complex relationships between these characters. Enhancing the model’s understanding and inference abilities regarding the structure of mathematical expressions, enabling it to correctly decode mathematical expressions that follow mathematical syntax rules, remains an important research question that the field continues to explore. The representation of mathematical expressions is mainly categorized into two types: sequence-based and tree-structure-based representations. The former is typified by \LaTeX expressions, which convert mathematical expressions into one-dimensional character sequences and use special structural control symbols (such as left curly braces “{” and right curly braces “}”) to express the hierarchical structure of the expressions. For example, the mathematical expression x^2 is represented in \LaTeX as

“ x^2 ”. The tree-structure-based representation, on the other hand, decomposes mathematical expressions into tuples, each tuple (p, c, r) comprising a parent node p , a child node c , and the relationship r between them. For instance, the same mathematical expression x^2 would be represented in this method as $(x, 2, \text{sup})$. Based on these two methods of representing mathematical expressions, the field of HMER has also developed two corresponding decoding methods: sequence-based decoding methods and tree-based decoding methods.

Sequence-based decoding methods target $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ sequences, which have the advantage of being a widely used method for expressing mathematical expressions. This compatibility not only aligns well with existing research in fields such as NLP and OCR but also offers strong versatility. However, this approach has its limitations. First, during the training process, the tree-like structural characteristics of mathematical expressions are not fully considered, which may lead to insufficient understanding of expressions structures by the model, thus affecting its generalization ability when dealing with complex structures. Second, as targets for sequence prediction, $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ sequences do not ensure the grammatical correctness of the model’s decoding results, especially when training on a relative small dataset, like CROHME dataset. For example, when dealing with complex mathematical expressions, the model might generate mismatched left and right curly braces, leading to predictions that do not conform to $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ syntax standards.

Tree-structure-based decoding methods treat mathematical expressions as tree-like structures for prediction and decoding. The advantage of this approach is that the model design fully integrates the characteristics of tree structures, theoretically enhancing the model’s ability to understand mathematical expressions and ensuring that it always decodes valid expressions during the prediction and inference process. Despite these benefits, tree-structure decoding models have their limitations. First, these models rely on RNN for construction, which cannot fully utilize the efficient parallel training features of Transformer models. Additionally, their decoding method is relatively complex and lacks the universality and broad applicability compared to $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ expressions. Most importantly, in practical experiments, the performance of tree-structure decoding models is often inferior to that of sequence decoding models, leading to a lack of widespread adoption of tree-structure decoding methods. Therefore, in practical engineering applications, engineers often prefer to choose sequence decoding models.

Addressing the respective advantages and drawbacks of these two decoding methods, in this work, we integrate the strengths of both sequence-based decoding and tree-based decoding models to propose a novel HMER model that is aware of tree structures of expressions, named TAMER (Tree-Aware Transformer). Our contributions are as follows:

(1) Without harming the flexibility and efficient training efficiency of sequence decoding models, TAMER introduces a novel Tree-Aware Module (TAM). It jointly optimizes both sequence prediction and tree structure prediction tasks, enhancing the model’s understanding and generalization capabilities for complex expression structures.

(2) In the inference phase, TAMER integrates a Tree Structure Prediction Scoring Mechanism into the beam search algorithm, allowing the model to consider the rationality of the expression tree structure while decoding $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ sequences. This mechanism ensures that the model’s recognition results are more complete and accurate in terms of grammatical structure.

(3) Experimental results across CROHME datasets demonstrate that the TAMER outperforms traditional sequence-based decoding and tree-based decoding methods, highlighting the potential of combining these two decoding methods. TAMER achieves expression recognition rate (ExpRate) of 61.23%, 60.26%, and 61.97% on the CROHME 2014/2016/2019 test sets (Mouchere et al. 2014; Mouchère et al. 2016; Mahdavi et al. 2019), respectively. Additionally, TAMER excels in avoiding syntactical errors, particularly in bracket matching, significantly outperforming other methods—especially on the CROHME datasets, which have relatively small training sets.

Related Work

HMER has long been a challenging research area within Optical Character Recognition (OCR) due to the nested hierarchical structures and diverse writing styles of handwritten expressions. Traditional HMER methods (Twaakyondo and Okamoto 1995; Ha, Haralick, and Phillips 1995; Alvaro, Sánchez, and Benedí 2014; Kosmala et al. 1999; Winkler 1996; Chan and Yeung 1998; Vuong, He, and Hui 2010; Chou 1989; Sakai 1961; Chan and Yeung 2000; Zanibbi, Blostein, and Cordy 2002; Toyota, Uchida, and Suzuki 2006) typically involve two steps: symbol recognition and structural analysis. Recently, deep learning methods (Zhang et al. 2017, 2020; Peng et al. 2021; Li et al. 2022; Wang et al. 2024) have become predominant. These methods commonly employ an encoder-decoder architecture. Depending on the difference of decoding manner, deep learning methods can be classified into sequence-based decoding methods and tree-based decoding methods.

Sequence-Based Decoding Methods

RNN-based Models In 2017, the WAP (Watch, Attend, and Parse) (Zhang et al. 2017) pioneered the use of deep neural networks for HMER, initiating significant advancements in this research field. Following WAP, DenseWAP (Zhang, Du, and Dai 2018) replace VGG encoder with DenseNet. Subsequent research has frequently adopted DenseNet as the primary backbone network for the visual encoder. PAL (Wu et al. 2019) and PALv2 (Wu et al. 2020) introduces an adversarial learning method utilizing handwritten-printed sample pairs. By learning semantic-invariant features, this approach enhances the accuracy of recognizing $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ expressions in various styles. The ABM (Bian et al. 2022) incorporates mutual learning loss between two directions to mitigate the issue of output imbalance. Recently, the CAN (Li et al. 2022) incorporates a Multi-Scale Counting Module that employs a symbolic counting task as an auxiliary task, which is jointly optimized alongside the expression sequence prediction task. All of these models employ decoders are based

on RNN. However, existing research (Bengio, Frasconi, and Simard 1993) indicates that models based on RNN have inherent limitations in modeling long-range dependencies between characters.

Transformer-based Models In the field of natural language processing, Transformer models have progressively supplanted traditional RNN architectures (He et al. 2016; Devlin et al. 2018). BTTR (Zhao et al. 2021) is the first to employ Transformer decoder for HMER and introduces a bidirectional training strategy to address the issue of output imbalance while fully leveraging bidirectional language information. CoMER (Zhao and Gao 2022) designs an Attention Refinement Module (ARM) that integrates coverage attention mechanism from RNN into the Transformer decoder, thereby alleviating the lack of coverage problem. Moreover, CoMER maintains the parallel training capabilities of the Transformer model and achieves impressive experimental results, making it an important backbone for subsequent methods. Based on CoMER and CAN, the GCN (Zhang et al. 2023) manually categorizes mathematical symbols into broader groups and utilizes the General Category Recognition Task as a supplementary task for joint optimization with the HMER task. The ICAL (Zhu, Gao, and Zhao 2024) introduces the Implicit Character Construction Module (ICCM) to model implicit character information within the $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ sequence. Additionally, it utilizes a Fusion Module to integrate ICCM outputs, thereby correcting the predictions of the Transformer Decoder.

Different from the autoregressive sequence-based decoding methods developed in recent years, NAMER (Liu et al. 2024) introduces a non-autoregressive modeling framework, which applies a parallel graph decoder to revise predicted visible symbols and establish connectivity between them, significantly reducing memory usage and inference costs.

Tree-Based Decoding Methods

DenseWAP-TD (Zhang et al. 2020) is the first work to employ a tree-based decoding model in the field of HMER, replacing the GRU decoder that directly regresses the $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ sequence with a decoder based on a two-dimensional tree structure. The TDv2 (Wu et al. 2022) is primarily optimized for the tree-based decoding technology in the DenseWAP-TD, with a particular focus on the diverse utilization of tree structures and the simplification of the decoding process. Furthermore, during the training process of the TDv2 model, various conversion methods are applied to the same $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ string. The Syntax-Aware Network (SAN) (Yuan et al. 2022) converts $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ sequences into a parsing tree, employing syntactic rules to reformulate the prediction of $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ sequences as a tree traversal task. In addition, SAN incorporates a novel Syntax-Aware Attention Module, effectively integrating syntactic information into the recognition process.

Method

Tree-Structure Annotation Construction

In constructing the annotated data for mathematical expression structure trees, we employ the method used by (Zhang

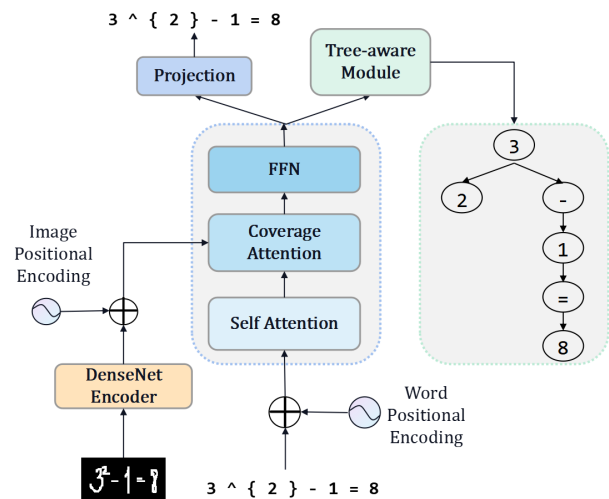


Figure 2: **The architecture of TAMER.** TAMER has 4 components: (1) Visual Encoder: DenseNet. (2) Sinusoidal Positional Encoding: image and word. (3) Decoder: Transformer Decoder with Coverage Attention. (4) Tree-Aware Module (TAM).

et al. 2020), which converts mathematical expressions based on $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ expressions into a tree structure. Specifically, the tree structure of a mathematical expression is represented as a series of tuples, each consisting of a child node c and its parent node p , denoted as (c, p) . To eliminate ambiguities introduced by repeated symbols and ensure compatibility with the $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ annotations used in the sequence-based decoding method, we utilize the index of mathematical symbols in the $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ sequences as node identifiers in the expression tree, rather than the symbols themselves. For instance, consider the mathematical expression depicted in Figure 2, with the corresponding $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ annotation “ $3 \wedge \{ 2 \} - 1 = 8$ ”. This expression can be annotated in its tree structure as “ $(0, -1), (1, -1), (2, -1), (3, 0), (4, -1), (5, 0), (6, 5), (7, 6), (8, 7)$ ”, where -1 signifies that a node lacks a parent and therefore does not contribute to the construction of the tree structure.

Model Architecture

The TAMER, as shown in the Figure 2, based on the CoMER (Zhao and Gao 2022), is divided into four main components: 1) The visual encoder extracts high-level visual features from input images of handwritten mathematical expressions, utilizing the DenseNet model (Huang et al. 2017)—widely applied in this recognition field—as its backbone network. 2) Image positional encoding and word positional encoding provide explicit positional information for visual features and word embedding vectors, respectively, using the same Sinusoidal Positional Encoding as in vanilla Transformer, BTTR and CoMER (Vaswani et al. 2017; Zhao et al. 2021; Zhao and Gao 2022). 3) The decoder, which employs a Transformer-based model adapted for this specific task, predicts the $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ sequence in an autoregressive

Tree Structure Prediction Scoring Mechanism

In addition to incorporating the loss from the tree structure prediction task during training to enhance the model’s understanding and generalization capabilities regarding mathematical expression structures, the Tree-Aware Module can also be used during the inference prediction phase to improve the structural coherence of the generated LaTeX sequences. This process includes the following three steps:

1. Use the beam search strategy to generate a set of candidate sequences, and calculate the sequence decoding score $S_{seq}(y)$ for each sequence y .

2. Utilize the Tree-Aware Module to compute the relationship score matrix for these sequences, thereby calculating the tree structure prediction score $S_{struct}(y)$ for each sequence.

3. Sum the sequence decoding score and the tree structure prediction score, i.e., $S_{seq}(y) + S_{struct}(y)$, and select the sequence with the highest composite score as the final output sequence.

By considering both sequence decoding and tree structure prediction scores, this method not only ensures the accuracy of the final generated sequence but also accounts for the rationality of its tree structure.

Experiments

Dataset

The CROHME Dataset, originating from the Online Handwritten Mathematical Expressions Recognition Competitions (CROHME) (Mouchere et al. 2014; Mouchère et al. 2016; Mahdavi et al. 2019) held over multiple years, is the preeminent benchmark for handwritten mathematical expression recognition. The training set comprises 8,836 handwritten mathematical expressions (HMEs), while the test sets from CROHME 2014 (Mouchere et al. 2014), 2016 (Mouchère et al. 2016), and 2019 (Mahdavi et al. 2019) contain 986, 1,147, and 1,199 HMEs, respectively. Each handwritten mathematical expression is stored in InkML format, capturing the trajectory coordinates of the handwritten strokes and providing the ground truth in both MathML and \LaTeX formats. For the purposes of model training and testing, the trajectory information from the InkML files is converted into grayscale bitmap images.

The HME100K dataset (Yuan et al. 2022) is a large-scale collection of real-scene handwritten mathematical expressions. It contains 74,502 training images and 24,607 testing images.

Evaluation Metrics

The Expression Recognition Rate (ExpRate) is the most widely employed evaluation metric for assessing the recognition of handwritten mathematical expressions. It is defined as the percentage of mathematical expressions correctly recognized out of the total number of expressions. In addition, we utilize the metrics “ ≤ 1 error” and “ ≤ 2 errors” to describe the performance of the model when up to one or two token prediction errors, respectively, are tolerated in the \LaTeX sequence.

Implementation Details

TAMER uses CoMER (Zhao and Gao 2022) as its baseline, employing a DenseNet (Huang et al. 2017) with the same hyperparameter configuration as the encoder, and the same Transformer (Vaswani et al. 2017) as the decoder. Further details can be found in the Appendix.

Comparison With State-of-the-Art Methods

Table 1 displays the performance of TAMER and previous methods on the CROHME dataset. For ease of comparison, in Table 1, we have roughly grouped the methods into RNN-based sequence decoding methods (group 1), tree-based decoding models (group 2), and Transformer-based methods (group 4). NAMER (group 3) (Liu et al. 2024), which utilizes a graph decoder and non-autoregressive decoding, and employs pre-trained DenseWAP (Zhang, Du, and Dai 2018) for assistance, is not categorized into the aforementioned groups due to its unique approach.

Since the data augmentation techniques used by some previous methods have not been disclosed, to ensure a fair comparison, the results shown in the table do not include any data augmentation. CoMER serves as the baseline for TAMER and is the backbone for many subsequent methods such as GCN (Zhang et al. 2023) and ICAL (Zhu, Gao, and Zhao 2024). However, the original CoMER publication does not provide results on the CROHME dataset without data augmentation. Therefore, we have reproduced the results of CoMER without data augmentation using their open-source code, indicated by † in Table 1. To ensure the robustness and reproducibility of our findings, we conducted experiments with both the baseline CoMER and the proposed TAMER using five different random seeds [7, 77, 777, 7777, 77777] under the same experimental conditions as ICAL (Zhu, Gao, and Zhao 2024). The reported results are the averages and standard deviations of these five experiments.

As shown in Table 1, TAMER consistently outperforms previous methods across all metrics. Notably, it significantly surpasses the baseline CoMER model by 2.85%, 3.28%, and 2.85% on the CROHME 2014/2016/2019 test sets, respectively. These experimental results clearly demonstrate the effectiveness of our approach.

Table 2 showcases the performance of TAMER on the HME100K dataset, where it exceeds the capabilities of both CoMER and NAMER across all assessed metrics. Compared to methods that typically incorporate auxiliary tasks, ICAL includes an additional Fusion Module, which effectively leverages auxiliary task information during training. Note that TAMER can also be easily extended with this kind of extra module. Consequently, we have included results for TAMER equipped with the Fusion Module. As indicated in the table, TAMER with the Fusion Module achieves SOTA performance on the HME100K dataset.

Performance Under Different Structural Complexity

To analyze in-depth the performance of TAMER on mathematical expressions of varying complexity, we first calculated the structural complexity of each test sample, and then

Method	CROHME 2014			CROHME 2016			CROHME 2019		
	ExpRate \uparrow	$\leq 1 \uparrow$	$\leq 2 \uparrow$	ExpRate \uparrow	$\leq 1 \uparrow$	$\leq 2 \uparrow$	ExpRate \uparrow	$\leq 1 \uparrow$	$\leq 2 \uparrow$
WAP	46.55	61.16	65.21	44.55	57.10	61.55	-	-	-
DenseWAP	50.1	-	-	47.5	-	-	-	-	-
DenseWAP-MSA	52.8	68.1	72.0	50.1	63.8	67.4	47.7	59.5	63.3
TAP	48.47	63.28	67.34	44.81	59.72	62.77	-	-	-
PAL	39.66	56.80	65.11	-	-	-	-	-	-
PAL-v2	48.88	64.50	69.78	49.61	64.08	70.27	-	-	-
WS-WAP	53.65	-	-	51.96	64.34	70.10	-	-	-
ABM	56.85	73.73	81.24	52.92	69.66	78.73	53.96	71.06	78.65
CAN-DWAP	57.00	74.21	80.61	56.06	71.49	79.51	54.88	71.98	79.40
CAN-ABM	57.26	74.52	82.03	56.15	72.71	80.30	55.96	72.73	80.57
DenseWAP-TD	49.1	64.2	67.8	48.5	62.3	65.3	51.4	66.1	69.1
TDv2	53.62	-	-	55.18	-	-	58.72	-	-
SAN	56.2	72.6	79.2	53.6	69.6	76.8	53.5	69.3	70.1
NAMER	60.51	75.03	82.25	60.24	73.5	80.21	61.72	75.31	82.07
BTTR	53.96	66.02	70.28	52.31	63.90	68.61	52.96	65.97	69.14
GCN	60.00	-	-	58.94	-	-	61.63	-	-
CoMER \dagger	58.38 ± 0.62	74.48 ± 1.41	81.14 ± 0.91	56.98 ± 1.41	74.44 ± 0.93	81.87 ± 0.73	59.12 ± 0.43	77.45 ± 0.70	83.87 ± 0.80
ICAL	60.63 ± 0.61	75.99 ± 0.77	82.80 ± 0.40	58.79 ± 0.73	76.06 ± 0.37	83.38 ± 0.16	60.51 ± 0.71	78.00 ± 0.66	84.63 ± 0.45
TAMER	61.23± 0.42	76.77± 0.78	83.25± 0.52	60.26± 0.78	76.91± 0.38	84.05± 0.41	61.97± 0.54	78.97± 0.42	85.80± 0.45

Table 1: **Performance comparison on the CROHME dataset.** We compare expression recognition rate (ExpRate) between our model and previous state-of-the-art models on the CROHME 2014/2016/2019 test sets. **None of the methods used data augmentation to ensure a fair comparison.** We denote our reproduced results with \dagger . All the performance results are reported in percentage (%).

Method	HME100K		
	ExpRate \uparrow	$\leq 1 \uparrow$	$\leq 2 \uparrow$
DenseWAP	61.85	70.63	77.14
DenseWAP-TD	62.60	79.05	85.67
ABM	65.93	81.16	87.86
SAN	67.1	-	-
CAN-DWAP	67.31	82.93	89.17
CAN-ABM	68.09	83.22	89.91
NAMER	68.52	83.10	89.30
BTTR	64.1	-	-
CoMER \dagger	68.12	84.20	89.71
ICAL	69.06	85.16	90.61
TAMER	68.52	84.61	89.94
TAMER w/ fusion	69.50	85.48	90.80

Table 2: **Performance comparison on the HME100K dataset(%).** We denote our reproduced results with \dagger . None of the methods used data augmentation to ensure a fair comparison.

assessed the ExpRate of TAMER when processing samples with different structural complexities. We define the structural complexity of a expression tree as the maximum number of nodes that have more than one child node across all search paths within the expression tree. Figure 4 provides examples of different mathematical expressions and their corresponding tree structural complexities. For instance, the expression $a + 1 = b$ is considered to have a complexity of 0 because all its nodes contain only one child node.

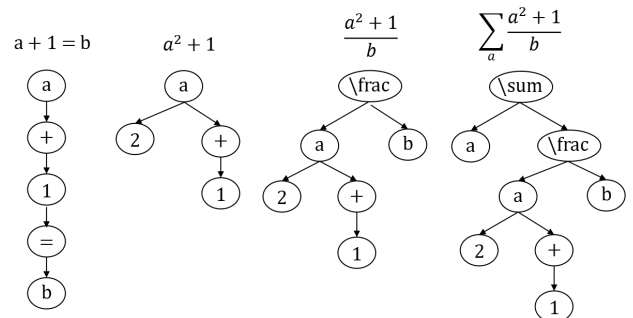


Figure 4: Examples of structural complexity for different expressions, with complexity ranging from 0 to 3.

Conversely, expressions containing mathematical structures like superscripts, fractions, and summation operators, which can include multiple child nodes, exhibit a higher tree structural complexity. As shown in Figure 5, we compares the baseline CoMER with TAMER, which incorporates a Tree-Aware Module, in terms of ExpRate on mathematical expressions of varying structural complexities within the CROHME 2014 dataset. The experimental results indicate that TAMER consistently achieves higher recognition accuracy than CoMER across all levels of structural complexity. Notably, TAMER’s recognition accuracy improves by 5.31% over CoMER when dealing with expressions having a structural complexity of 5 or higher, marking the most significant improvement among all compared complexity levels.

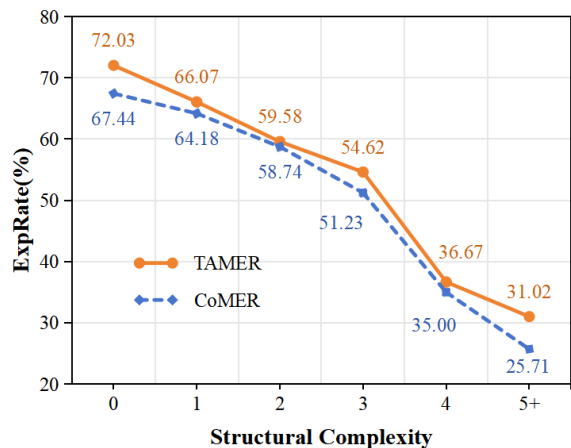


Figure 5: ExpRate under different structural complexities on CROHME 2014(in %).

Bracket matching issues are common syntactical errors in sequence-based decoding methods. We also compared TAMER and the other two models on the CROHME 2014 dataset in terms of bracket matching accuracy at different structural complexities, as shown in Figure 1. The figure shows that TAMER consistently maintains an accuracy rate of over 92% across various structural complexities. Particularly for expressions with a complexity level exceeding 5, TAMER significantly surpasses CoMER by more than 15%. Additionally, although TAMER slightly underperforms ICAL in ExpRate on the HME100K dataset, it significantly surpasses ICAL in terms of syntactical accuracy of the generated expressions, especially in Bracket Matching Accuracy as shown in Figure 6. Such experimental results demonstrate that TAMER, which is jointly optimized with tree structure prediction tasks in addition to sequence decoding tasks, exhibits stronger generalization capabilities in handling complex structured mathematical expressions, thus validating its effectiveness in processing complex expressions.

Ablation Study

We performed an ablation study on the CROHME 2014/2016/2019 test sets to assess the impact of the Tree-Aware Module and Tree Structure Prediction Scoring Mechanism on our TAMER model’s performance. Results summarized in Table 3 show that models utilizing both the Tree-Aware Module and Tree Scoring consistently outperform configurations without these components. Specifically, on the CROHME 2014 test set, the full TAMER model achieved an expression recognition rate (ExpRate) of 61.23%, compared to 60.39% when only the Tree Aware module was used, and 58.38% for the baseline without any modules. This pattern of enhanced performance with the inclusion of both modules is consistent across the 2016 and 2019 test sets.

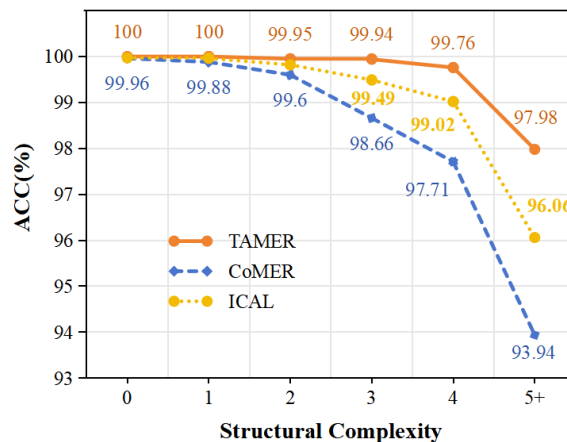


Figure 6: Bracket Matching Accuracy under different structural complexities on HME100K(in %).

Dataset	TAM	Tree Scoring	ExpRate
CROHME 2014	✓		58.38
	✓	✓	61.23
CROHME 2016	✓		56.98
	✓	✓	60.26
CROHME 2019	✓		59.12
	✓	✓	61.97

Table 3: Ablation study on the CROHME 2014/2016/2019 test sets(in %). The terms "TAM" and "Tree Scoring" indicate whether the Tree-Aware Module or the Tree Structure Prediction Scoring Mechanism was utilized, respectively.

Conclusion

In this paper, we introduce TAMER, which significantly enhances the understanding and generalization of complex mathematical expression structures by integrating sequence-based and tree-based decoding methods. Our main contributions are threefold: (1) We present a new Tree-Aware Module that allows the model to perform sequence decoding tasks and predict tree structures simultaneously. (2) In the inference phase, we incorporate a Tree Structure Prediction Scoring Mechanism into the beam search algorithm. (3) Experimental results demonstrate that the TAMER surpasses previous state-of-the-art approaches on the CROHME 2014, 2016, and 2019 datasets, achieving ExpRate of 61.23%, 60.26%, and 61.97% , respectively. Additionally, TAMER excels at avoiding syntactical errors, particularly in bracket matching, where it significantly outperforms other methods. This superiority is especially notable on the CROHME datasets, which have relatively small training sets.

Acknowledgements

This work is supported by the projects of National Natural Science Foundation of China (No. 62376012) and Beijing Science and Technology Program (Z231100007423011), which is also a research achievement of State Key Laboratory of Multimedia Information Processing and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology).

References

- Alvaro, F.; Sánchez, J.-A.; and Benedí, J.-M. 2014. Recognition of on-line handwritten mathematical expressions using 2D stochastic context-free grammars and hidden Markov models. *Pattern Recognition Letters*, 35: 58–67.
- Bengio, Y.; Frasconi, P.; and Simard, P. 1993. The problem of learning long-term dependencies in recurrent networks. In *IEEE international conference on neural networks*, 1183–1188.
- Bian, X.; Qin, B.; Xin, X.; Li, J.; Su, X.; and Wang, Y. 2022. Handwritten mathematical expression recognition via attention aggregation based bi-directional mutual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 113–121.
- Chan, K.-F.; and Yeung, D.-Y. 1998. Elastic structural matching for online handwritten alphanumeric character recognition. In *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*, volume 2, 1508–1511. IEEE.
- Chan, K.-F.; and Yeung, D.-Y. 2000. An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions. *Pattern recognition*, 33(3): 375–384.
- Chou, P. A. 1989. Recognition of equations using a two-dimensional stochastic context-free grammar. In *Visual Communications and Image Processing IV*, volume 1199, 852–865. SPIE.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Ha, J.; Haralick, R. M.; and Phillips, I. T. 1995. Understanding mathematical expressions from document images. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 2, 956–959. IEEE.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.
- Kosmala, A.; Rigoll, G.; Lavitrotte, S.; and Pottier, L. 1999. On-line handwritten formula recognition using hidden Markov models and context dependent graph grammars. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*, 107–110. IEEE.
- Li, B.; Yuan, Y.; Liang, D.; Liu, X.; Ji, Z.; Bai, J.; Liu, W.; and Bai, X. 2022. When counting meets HMER: counting-aware network for handwritten mathematical expression recognition. In *European Conference on Computer Vision*, 197–214. Springer.
- Liu, C.; Pan, J.; Hu, J.; Yin, B.; Yin, B.; Chen, M.; Liu, C.; Du, J.; and Liu, Q. 2024. NAMER: Non-Autoregressive Modeling for Handwritten Mathematical Expression Recognition. *arXiv preprint arXiv:2407.11380*.
- Mahdavi, M.; Zanibbi, R.; Mouchere, H.; Viard-Gaudin, C.; and Garain, U. 2019. ICDAR 2019 CROHME+ TFD: Competition on recognition of handwritten mathematical expressions and typeset formula detection. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1533–1538.
- Mouchere, H.; Viard-Gaudin, C.; Zanibbi, R.; and Garain, U. 2014. Icfhr 2014 competition on recognition of on-line handwritten mathematical expressions (crohme 2014). In *2014 14th International Conference on Frontiers in Handwriting Recognition*, 791–796.
- Mouchère, H.; Viard-Gaudin, C.; Zanibbi, R.; and Garain, U. 2016. ICFHR2016 CROHME: Competition on recognition of online handwritten mathematical expressions. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 607–612.
- Peng, S.; Gao, L.; Yuan, K.; and Tang, Z. 2021. Image to LaTeX with graph neural network for mathematical formula recognition. In *Document Analysis and Recognition-ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II 16*, 648–663. Springer.
- Sakai, I. 1961. Syntax in universal translation. In *Proceedings of the International Conference on Machine Translation and Applied Language Analysis*.
- Toyota, S.; Uchida, S.; and Suzuki, M. 2006. Structural analysis of mathematical formulae with verification based on formula description grammar. In *Document Analysis Systems VII: 7th International Workshop, DAS 2006, Nelson, New Zealand, February 13-15, 2006. Proceedings 7*, 153–163. Springer.
- Twaakyondo, H.; and Okamoto, M. 1995. Structure analysis and recognition of mathematical expressions. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, 430–437 vol.1.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30: 5998–6008.
- Vuong, B.-Q.; He, Y.; and Hui, S. C. 2010. Towards a web-based progressive handwriting recognition environment for mathematical problem solving. *Expert Systems with Applications*, 37(1): 886–893.
- Wang, J.; Hu, K.; Zhong, Z.; Sun, L.; and Huo, Q. 2024. Detect-Order-Construct: A Tree Construction based Approach for Hierarchical Document Structure Analysis. *arXiv preprint arXiv:2401.11874*.

- Winkler, H.-J. 1996. HMM-based handwritten symbol recognition using on-line and off-line features. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 6, 3438–3441. IEEE.
- Wu, C.; Du, J.; Li, Y.; Zhang, J.; Yang, C.; Ren, B.; and Hu, Y. 2022. TDv2: A Novel Tree-Structured Decoder for Offline Mathematical Expression Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2694–2702.
- Wu, J.-W.; Yin, F.; Zhang, Y.-M.; Zhang, X.-Y.; and Liu, C.-L. 2019. Image-to-markup generation via paired adversarial learning. In *Machine learning and knowledge discovery in databases: European conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, 18–34. Springer.
- Wu, J.-W.; Yin, F.; Zhang, Y.-M.; Zhang, X.-Y.; and Liu, C.-L. 2020. Handwritten mathematical expression recognition via paired adversarial learning. *International Journal of Computer Vision*, 128: 2386–2401.
- Yuan, Y.; Liu, X.; Dikubab, W.; Liu, H.; Ji, Z.; Wu, Z.; and Bai, X. 2022. Syntax-aware network for handwritten mathematical expression recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4553–4562.
- Zanibbi, R.; Blostein, D.; and Cordy, J. R. 2002. Recognizing mathematical expressions using tree transformation. *IEEE Transactions on pattern analysis and machine intelligence*, 24(11): 1455–1467.
- Zhang, J.; Du, J.; and Dai, L. 2018. Multi-scale attention with dense encoder for handwritten mathematical expression recognition. In *2018 24th international conference on pattern recognition (ICPR)*, 2245–2250.
- Zhang, J.; Du, J.; Yang, Y.; Song, Y.-Z.; Wei, S.; and Dai, L. 2020. A Tree-Structured Decoder for Image-to-Markup Generation. In *ICML*, In Press.
- Zhang, J.; Du, J.; Zhang, S.; Liu, D.; Hu, Y.; Hu, J.; Wei, S.; and Dai, L. 2017. Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition. *Pattern Recognition*, 71: 196–206.
- Zhang, X.; Ying, H.; Tao, Y.; Xing, Y.; and Feng, G. 2023. General Category Network: Handwritten Mathematical Expression Recognition with Coarse-Grained Recognition Task. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.
- Zhao, W.; and Gao, L. 2022. CoMER: Modeling Coverage for Transformer-Based Handwritten Mathematical Expression Recognition. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, 392–408. Springer.
- Zhao, W.; Gao, L.; Yan, Z.; Peng, S.; Du, L.; and Zhang, Z. 2021. Handwritten mathematical expression recognition with bidirectionally trained transformer. In *Document Analysis and Recognition—ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II 16*, 570–584. Springer.
- Zhu, J.; Gao, L.; and Zhao, W. 2024. ICAL: Implicit Character-Aided Learning for Enhanced Handwritten Mathematical Expression Recognition. *arXiv preprint arXiv:2405.09032*.